



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/737,527	12/13/2000	Paul F. Austin	5150-47600	9833

7590

06/04/2003

Jeffrey C. Hood
Conley, Rose & Tayon, P.C.
P. O. Box 398
Austin, TX 78767-0398

EXAMINER

BASOM, BLAINE T

ART UNIT

PAPER NUMBER

2173

DATE MAILED: 06/04/2003

3

Please find below and/or attached an Office communication concerning this application or proceeding.

PD

Office Action Summary

Application No.

09/737,527

Applicant(s)

AUSTIN ET AL.

Examiner

Blaine Basom

Art Unit

2173

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☐ Responsive to communication(s) filed on ____.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-68 is/are pending in the application.
- 4a) Of the above claim(s) ____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) ____ is/are allowed.
- 6) ☒ Claim(s) 1-68 is/are rejected.
- 7) ☐ Claim(s) ____ is/are objected to.
- 8) ☐ Claim(s) ____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 13 December 2000 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- 11) ☐ The proposed drawing correction filed on ____ is: a) ☐ approved b) ☐ disapproved by the Examiner.
- If approved, corrected drawings are required in reply to this Office action.
- 12) ☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

- 13) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. ____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
- a) ☐ The translation of the foreign language provisional application has been received.
- 15) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892) 4) ☐ Interview Summary (PTO-413) Paper No(s). ____
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948) 5) ☐ Notice of Informal Patent Application (PTO-152)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) ____ 6) ☐ Other:

DETAILED ACTION***Information Disclosure Statement***

The information disclosure statement filed 3/29/2001 fails to comply with the provisions of 37 CFR 1.97, 1.98 and MPEP § 609 because the references cited therein do not constitute prior art. It has been placed in the application file, but the information referred to therein has not been considered as to the merits. Applicant is advised that the date of any re-submission of any item of information contained in this information disclosure statement or the submission of any missing element(s) will be the date of submission for purposes of determining compliance with the requirements based on the time of filing the statement, including all certification requirements for statements under 37 CFR 1.97(e). See MPEP § 609 ¶ C(1).

Double Patenting

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and, *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

Claim 1 is provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claim 1 of copending Application No.

Art Unit: 2173

09/737639. Although the conflicting claims are not identical, they are not patentably distinct from each other. In the instant application, claim 1 recites a method for configuring a graphical user interface (GUI) element, the method comprising: displaying a GUI element on a display, receiving user input specifying at least one of a data source or data target, and automatically configuring the GUI element to perform at least one of: receiving data from the specified data source, and/or publishing data to the specified data target. Similarly, claim 1 of copending Application No. 09/737639 recites a method for configuring a GUI element, the method comprising receiving user input specifying at least one of a data source or data target, automatically displaying a GUI element in response to the user input, and automatically configuring the GUI element to perform at least one of: receiving data from the specified data source, and/or publishing data to the specified data target. Consequently, claim 1 of copending Application No. 09/737639 differs from claim 1 of the instant application in that claim 1 of the copending application expresses displaying a GUI element in response to the user input, whereas claim 1 of the present application expresses no such idea. However, since claim 1 of the instant application simply expresses "displaying a GUI element on a display," it is understood that within the scope of this claim, the GUI element may be displayed in response to the user input. Thus claim 1 of copending Application No. 09/737639 falls entirely within the scope of claim 1 of the instant application, or in other words, claim 1 is anticipated by claim 1 of copending Application No. 09/737639.

By similar reasoning, claims 22, 28, and 55 are provisionally rejected as being unpatentable over claim 1 of copending Application No. 09/737639, claims 33 and 43 are provisionally rejected as being unpatentable over claim 12 of copending Application No.

Art Unit: 2173

09/737639, claim 56, 61, and 67 are provisionally rejected as being unpatentable over claim 18 of copending Application No. 09/737639, claim 60 is provisionally rejected as being unpatentable over claim 4 of copending Application No. 09/737639, claim 62 is provisionally rejected as being unpatentable over claim 34 of copending Application No. 09/737639, and claim 68 is provisionally rejected as being unpatentable over claim 38 of copending Application No. 09/737639.

Furthermore, as the ideas expressed by dependent claims 2-21, 23-27, 29-32, 34-42, 43-54, 57-59, and 63-66 of the present application are each similarly expressed by one or more of claims 1-42 of copending Application No. 09/737639, claims 2-21, 23-27, 29-32, 34-42, 43-54, 57-59, and 63-66 of the present application are provisionally rejected as being unpatentable over copending Application No. 09/737639.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-11, 12-14, 20-27, 33-39, and 60-68 are rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 5,339,392, which is attributed to Risberg et al. (and hereafter referred to as “Risberg”). In general, Risberg discloses an application to be used for monitoring and managing complex systems having a plurality of frequently varying data values. More specifically, and regarding the claimed invention, this application allows users to create custom graphical user interfaces in which these data values are displayed, and in which changes in these data values are immediately reflected on the display (see column 1, lines 31-41). These data values, which are received from one or more sources over a network, are particularly displayed via one or more GUI elements. It is thus understood that Risberg teaches a method for configuring a GUI element to publish or subscribe to a data target or data source.

As per claim 1, the application disclosed by Risberg is used to construct GUIs for viewing financial information such as stock prices. Risberg however notes that the application also applies to *any* system which generates real time data that must be monitored (see column 2, lines 52-55). In any event, Risberg discloses that the data to be monitored is displayed by a plurality of GUI elements, specifically “quotes,” “dynamic graphs,” “tickers,” or “page fragments” (see column 28, lines 30-67). For example, a page fragment displays a section of

Art Unit: 2173

data obtained from a financial data source. The data displayed via a fragment element is updated in real time (see column 28, lines 62-67). Quotes, dynamic graphs, and tickers similarly display data from one or more financial sources, except in a different format. To create a page fragment on the GUI, a user uses a "Page Fragment tool" and drags, with a mouse, a region on the GUI where the page fragment is to be positioned. In response, the page fragment is displayed but contains no information (see column 11, line 65 – column 12, line 4). For the page fragment to display information, the user enters a data source, i.e. "service," into a specific dialog box provided to the user, wherein the data service provides financial data which is displayed in the page fragment (see column 12, lines 5-28). As this data displayed by the page fragment is updated in real time (see column 28, lines 62-67), it is understood that the page fragment is thus configured to receive and display data from the specified data source. It is interpreted that quotes, dynamic graphs, and tickers are created and configured by similar means. Lastly, Risberg notes that the data displayed by a quote, dynamic graph, ticker, or page fragment can be published on a network so that it may be used as a bulletin board or by other users linked to the network (see column 3, line 66 – column 4, line 4). Thus regarding claim 1, Risberg teaches displaying a GUI element, such as a page fragment, on a display; receiving user input, specifically through a dialog box, wherein this user input specifies at least one of a data source or data target with which to associate the GUI element; and, in response to receiving this input, automatically configuring the GUI element to perform at least one of: receiving and displaying data from the specified data source; and/or publishing data associated with the GUI element to the specified data target.

Art Unit: 2173

In reference to claims 2-5, the page fragment disclosed by Risberg is configured by entering a particular data source into a dialog box provided to the user, whereby as shown above, the page fragment resultantly displays data received from this data source (see column 12, lines 5-28). It is interpreted that the other graphical elements disclosed by Risberg are configured by similar means. Thus these GUI elements are automatically configured without user programming and without the user input specifying source code. Particularly, they are configured in response to receiving user input via a user interface dialog box. After these GUI elements are configured, they receive and display data from the specified source in real time.

With respect to claim 6, the application disclosed by Risberg, which as described above is used to construct GUIs for viewing financial information, is implemented on a first computer which accesses over a network a second computer, specifically a server. The application receives data from the server and displays the data via one or more graphical elements, such as page fragments, quotes, tickers, or dynamic graphs (see column 2, lines 27-48). Thus it is understood that the method taught by Risberg is executed on a first computer, whereby this first computer is operable to connect to a second computer over a network. It is further understood that this second computer comprises a data source, whereby the GUI elements taught by Risberg may be configured to connect to the second computer and receive and display data from this data source.

Referring to claims 7-9, the application disclosed by Risberg is used to construct GUIs, or more specifically "active documents." Each active document comprises a set of "sheets," considered a GUI, which is created by the user. In addition, each active document may comprise one or more scripts which define the functionality of various components of the active document

Art Unit: 2173

(see column 2, lines 27-39). In any event, an active document is considered a computer program, as it is executed on a computer to access data over a network and display it on the computer.

Thus the page fragments, quotes, dynamic graphs, and tickers described by Risberg are GUI elements associated with a first computer program, specifically an active document. These GUI elements are included in a user interface associated with an active document (for example, see figure 1, which shows the user interface of an active document; a quote, which is designated by reference number 18; a ticker, which is designated by reference number 20; a dynamic graph, which is designated by reference number 22; and a page fragment, which is designated by reference number 24, are displayed in this user interface). As shown above, these GUI elements are operable to receive and display data from a specific data source during execution of the active document. Risberg notes that the specification of such a data source may be received during the development the active document (for example, see column 11, line 65 – column 12, line 28).

In regard to claim 10, an active document is considered a graphical program. For example, figure 1 shows the user interface of an active document. A quote, which is designated by reference number 18; a ticker, which is designated by reference number 20; a dynamic graph, which is designated by reference number 22; and a page fragment, which is designated by reference number 24, are all elements which present data graphically. In the case of quotes, page fragments, and tickers, data is presented in lists or charts. With dynamic graphs, data is presented in a graph. Moreover, the interface of the active document may contain other graphical elements to actuate specific functionality of the program (see column 4, lines 4-29).

Art Unit: 2173

With respect to claim 11, the application disclosed by Risberg, which as described above is used to construct GUIs for viewing financial information, is implemented on a first computer which accesses over a network a second computer, specifically a server. The application receives data from the server and displays the data via one or more graphical elements, such as page fragments, quotes, tickers, or dynamic graphs (see column 2, lines 27-48). Consequently, it is understood that the data source may be from the group consisting of an HTTP server, an FTP server, and OPC server, an SNMP server, and a DataSocket server, which are all well-known types of servers in the art.

In reference to claims 12 and 13, Risberg discloses that for a page fragment to display information, the user enters a data source into a specific dialog box provided to the user, wherein this data source provides financial data which is displayed by the page fragment (see column 12, lines 5-28). As this data displayed by the page fragment is updated in real time (see column 28, lines 62-67), it is understood that the page fragment is thus configured to receive and display data from the specified data source. It is interpreted that quotes, dynamic graphs, and tickers are created and configured by similar means. Risberg therefore teaches providing user input specifying a data source with which to associate a GUI element, wherein when automatically configuring the GUI element, the GUI element is configured to receive and display data from this data source. Specifically regarding claim 13, Risberg further discloses that this data source is a remote data source associated with a remote computer, specifically a server (see column 2, lines 27-48). Thus it is understood that the GUI elements, which receive and display this data, are configured to connect to this remote data source and receive and display data from this remote data source during program execution.

With respect to claim 14, the application presented by Risberg, as is taught above, teaches a method for enabling reader programs, i.e. active documents, to display the data generated by a source. Specifically, the GUI elements disclosed by Risberg, which are presented in active documents, are each configured by entering a particular data source into a dialog box provided to the user, whereby as shown above, the particular GUI element resultantly displays data received from this data source. This data source is a server program located over a network (see column 21, lines 49-58). Risberg further notes that the GUI element may display data from "a calculation published from [a] RealTime spreadsheet or a C program" (see column 28, lines 40-47). Risberg thus teaches executing a computer program, i.e. a RealTime spreadsheet or C program, which is understood to be operable to publish live data to the server.

Regarding claims 20 and 21, the data received and displayed by the GUI elements disclosed by Risberg is displayed in real time (for example, see column 2, lines 39-48). Consequently, the data is considered live data. Moreover, the data is financial data, which is obtained or measured by one of a plurality of services, such as "MarketFeed 2," "Telerate," or "Reuters" (for example, see column 28, lines 40-47). The data is therefore also considered measurement data.

The ideas expressed in claims 22-24 are also expressed in claims 1-3, respectively. Consequently, claims 22-24 are rejected for the reasons cited above in the rejection for claims 1-3.

With respect to claim 25, the application disclosed by Risberg is used to construct GUIs, wherein as described above, one or more GUI elements are configured to receive and indicate data from a specific data source. For example, figure 1 presents an illustration of such a GUI.

Reference number 24 denotes a page fragment, which as shown above, receives and indicates a segment of financial data. As shown in figure 1, this financial data is particularly displayed to the user. Thus Risberg teaches indicating the received data by displaying it.

The ideas expressed in claim 26 are also expressed in claim 7. Consequently, claim 26 is rejected for the reasons cited above in the rejection for claim 7.

Regarding claim 27, Risberg discloses that for a page fragment to display information, the user enters a data source into a specific dialog box provided to the user, wherein this data source provides financial data which is displayed in the page fragment (see column 12, lines 5-28). As this data displayed by the page fragment is updated in real time (see column 28, lines 62-67), it is understood that the page fragment is thus configured to receive and display data from the specified data source. It is interpreted that quotes, dynamic graphs, and tickers are configured by similar means. In any event, it is understood that the user input specifying the data source does not specify any source code for the computer program.

In reference to claim 33, the application disclosed by Risberg involves displaying a GUI element associated with a first program, specifically an active document, wherein the GUI element is displayed in response to user input received during development of the active document, as was shown above in the rejection for claim 7. Moreover, as shown above in the rejection for claim 1, Risberg teaches receiving user input during development of the active document, the user input specifying a data source with which to associate the GUI elements, and automatically configuring the active document to receive data from the specified data source and display the data in the GUI element during program execution, in response to receiving the user input specifying the data source.

The ideas expressed in claim 34 are also expressed in claim 27. Consequently, claim 34 is rejected for the reasons cited above in the rejection for claim 27. Similarly, because the ideas expressed in claims 35 and 36 are also expressed in claims 9 and 10, respectively, claims 35 and 36 are rejected for the reasons cited above in the rejections for claims 9 and 10. Likewise, claim 37 is rejected for the reasons cited above in the rejection for claim 6, since the ideas expressed in claim 37 are also expressed in rejected claim 6.

In regard to claim 38, the GUI elements disclosed by Risberg are each configured by entering a particular data source into a dialog box provided to the user, whereby as shown above, the particular GUI element resultantly displays data received from this data source. This data source is a server program located over a network (see column 21, lines 49-58). Risberg further notes that the GUI element may display data from "a calculation published from [a] RealTime spreadsheet or a C program" (see column 28, lines 40-47). Thus Risberg teaches creating a second computer program, namely a RealTime spreadsheet or C program, which is operable to publish live data to a remote data source, and wherein an active document is operable to display the live data published by the second computer program in a GUI element.

In regard to claim 39, the GUI elements disclosed by Risberg are each configured by entering a particular data source into a dialog box provided to the user, whereby as shown above, the particular GUI element resultantly displays data received from this data source. This data source is a server program located over a network (see column 21, lines 49-58).

As per claim 60, Risberg teaches a method for enabling a graphical program to exchange data with a data source or target, as is described above. As particularly shown above in the rejection for claim 1, this method comprises displaying a GUI element in a graphical program

Art Unit: 2173

and configuring this GUI element with a data connection to a data source or target. Risberg further teaches that configuring this GUI element does not necessitate specifying source code for the graphical program, as is shown above in the rejection for claim 3.

With respect to claim 61, the application presented by Risberg, as is taught above, teaches a method for enabling multiple reader programs to display the data generated by a writer program. Specifically, the GUI elements disclosed by Risberg are each configured by entering a particular data source into a dialog box provided to the user, whereby as shown above, the particular GUI element resultantly displays data received from this data source. This data source is a server program located over a network (see column 21, lines 49-58). Risberg further notes that the GUI element may display data from "a calculation published from [a] RealTime spreadsheet or a C program" (see column 28, lines 40-47). Thus Risberg teaches creating and executing a writer program, namely a RealTime spreadsheet or a C program, which is operable to publish data to a server, and also, creating a reader program, specifically an active document, which as described above has a GUI element configured to receive the data published to the server and display the data. As shown above, creating the reader program comprises automatically configuring this GUI element to receive and display the data published to the server. Risberg further notes that the active documents created by the application, comprises a GUI, denoted by "sheets," which may be used throughout a department or firm (see column 26, lines 53-65). It is therefore understood that such an active document may be accessed by more than one user in the firm, and more specifically, may be displayed by more than one computer. Consequently, Risberg teaches executing the active document, i.e. reader program, on a plurality

Art Unit: 2173

of computers, where on each computer the reader program is operable to receive the data published by the server and display the data in a GUI element.

Referring to claim 62, the method taught by Risberg, which is described above in the rejection for claim 1, is implemented as a software program (for example, see column 26, lines 53-68). Consequently, it is understood that it is executed by a computer system, the computer system having display device, a processor, and a memory coupled to the processor. Such a computer system implementing the method taught by Risberg is considered a system, like that recited in claim 62, which is for configuring a GUI element to publish or subscribe to a data target or source.

The ideas expressed in claims 63 and 64 are also expressed in claims 2 and 6, respectively. Consequently, claims 63 and 64 are rejected for the reasons cited above in the rejection for claims 2 and 6.

As per claims 65 and 66, the application taught by Risberg is executed to create an active document, i.e. a second program (see column 2, lines 27-31). Thus it is understood that the processor implementing the application taught by Risberg is operable to receive user input for creating a second program, wherein a GUI element is associated with the second program, and more specifically, wherein the GUI element is included in the user interface of the second program. Moreover, this user input specifies at least one of a data source or data target (for example, see column 11, line 35 – column 12, line 28, which describe the operations required for adding a GUI element, specifically a page fragment, to the user interface of an active document). As the data displayed by the GUI element is updated in real time (see column 28, lines 62-67), it

Art Unit: 2173

is understood that the page fragment is thus automatically configured to receive and display data from the specified data source during execution of the active document.

With respect to claim 67, the application presented by Risberg, as is taught above, teaches a method for enabling reader programs, i.e. active documents, to display the data generated by a writer program. Specifically, the GUI elements disclosed by Risberg, which are presented in active documents, are each configured by entering a particular data source into a dialog box provided to the user, whereby as shown above, the particular GUI element resultantly displays data received from this data source. This data source is a server program located over a network (see column 21, lines 49-58). Risberg further notes that the GUI element may display data from "a calculation published from [a] RealTime spreadsheet or a C program" (see column 28, lines 40-47). Risberg thus teaches a writer program, i.e. a RealTime spreadsheet or C program, which is understood to be executable to write data to the server. It is interpreted that the server, active document, and writer program execute on separate computer systems, as is common in the art. Thus Risberg teaches a system for exchanging data, the system comprising a first computer system coupled to a second computer system via a network, whereas common in the art, the first computer system and the second computer system each include a memory medium coupled to a processor. Moreover, a third computer system is coupled to the second computer system via a network wherein the third computer system includes a memory medium coupled to a processor; and wherein a writer program is stored in the memory of the first computer system; a server program is stored in the memory of the second computer system; and a reader program is stored in the memory of the third computer system. The processor of the first computer system is operable to execute the writer program to write data to the server

Art Unit: 2173

program; and the processor of the third computer system is operable to execute the reader program to receive the data from the server program and display the data in a GUI element, wherein the reader program does not include user-specified source code for receiving and displaying the data in the GUI element.

Referring to claim 68, the method taught by Risberg, which is described above in the rejection for claim 1, is implemented as a software program (for example, see column 26, lines 53-68). Consequently, it is understood that it is executed by a computer system, the computer system having a memory upon which the program is stored. Such a computer memory storing the software program disclosed by Risberg is considered a memory medium, like that recited in claim 68, which is for configuring a GUI element to publish or subscribe to a data target or source.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 15-19, 28-32, and 40-59 are rejected under 35 U.S.C. 103(a) as being obvious over the teachings of Risberg, which is described above, and also over U.S. Patent No. 5,959,621, which is attributed to Nawaz et al. (and hereafter referred to as "Nawaz"). Specifically regarding claim 15, Risberg teaches a method for configuring a GUI element, wherein as shown above in the rejection for claim 1, the method comprises displaying a GUI element on a display, and receiving user input specifying a data source with which to associate the GUI element. In response, the GUI element is automatically configured to receive and display data from the specified data source. Moreover, Risberg teaches that the data displayed in a constructed active document, which is interpreted to be the data displayed via such a GUI element, may be published on a network (see column 3, line 66 – column 6, line 4). Thus it is understood that Risberg discloses configuring a GUI element to publish data. However, and with respect to the claimed invention, Risberg does not specifically teach the steps undertaken to publish such data to a specific target. In other words, Risberg does not explicitly teach receiving user input specifying a first data target with which to associate the GUI element, and

automatically configuring the GUI element to publish data associated with the GUI element to this first data target, as is expressed in claim 15.

Like Risberg, Nawaz presents a method for configuring a GUI element, specifically a ticker, to receive and display data from a specific data source. With further similarity to the teachings of Risberg, this data may comprise financial data, namely stock prices (see column 3, lines 27-29). With respect to the claimed invention, the ticker disclosed by Nawaz, like the GUI elements of Risberg, may be configured to publish data associated with the ticker to one or more specific targets. Specifically, a user uses a "posting page" associated with the ticker to determine which users, i.e. targets, receive data associated with the ticker (see column 11, lines 40-55).

It would have therefore been obvious to one of ordinary skill in the art, having the teachings of Risberg and Nawaz before him at the time the invention was made, to modify the method taught by Risberg such that a page similar to the posting page of Nawaz is utilized to configure a GUI element to publish data to a specific data target. In other words, it would have been obvious to modify the method of Risberg such that, with a posting page, a user specifies a first data target with which to associate the GUI element, and consequently, the GUI element is automatically configured to publish data associated with the GUI element to this data target. One would have been motivated to create such a combination because, as is demonstrated by Nawaz, a posting page allows a user to efficiently delineate who should and should not receive data associated with a GUI element. The provision of such a utility is beneficial when publishing data, as is shown by Nawaz.

In reference to claim 16, the combination of Risberg and Nawaz described above teaches a posting page, wherein a user enters one or more data targets to publish information associated

Art Unit: 2173

with a GUI element. It is understood that these data targets are remote data targets, each associated with a remote computer. Thus it is also understood that the GUI element is automatically configured to connect to each remote data target and publish data associated with the GUI element to the remote data target.

Referring to claim 17, the combination of Risberg and Nawaz described above teaches configuring a GUI element to publish data to a remote data target. Nawaz teaches that a GUI element at the target, specifically a ticker, may be configured to receive and display this published data (see column 10, lines 16-28). It is therefore understood that the combination of Risberg and Nawaz described above teaches executing a computer program at the remote target, wherein the program is operable to receive data from the remote data target, and wherein the computer program is operable to display the data.

As per claims 18 and 19, Risberg teaches receiving user input specifying a data source with which to associate a GUI element, wherein as described above, the GUI element is automatically configured to receive and display data from this specified data source. Similarly, the combination of Risberg and Nawaz further teaches receiving user input specifying a data target with which to associate the GUI element, wherein as described above, the GUI element is automatically configured to publish data associated with the GUI element to the specified data target. It is understood that a user may specify both a data source and a data target with which to associate a GUI element. Moreover, it is understood that the specified data source may be the same as the specified data target. In such a case, the GUI element would be automatically configured to receive and display data from the remote data source, and publish data to the specified data target.

The ideas expressed in claim 28 are also expressed in claim 15. Consequently, claim 28 is rejected for the reasons cited above in the rejection for claims 15.

In regard to claim 31, the application disclosed by Risberg is used to construct GUIs, or more specifically, active documents. As described above, each active document comprises a set of sheets, considered a GUI, which is created by the user. In addition, each active document may comprise one or more scripts which define the functionality of various components of the active document (see column 2, lines 27-39). In any event, an active document is considered a computer program, as it is executed on a computer to access data over a network and display it to the user. The GUI elements taught by the combination of Risberg and Nawaz are thus associated with a first computer program, specifically an active document. These GUI elements are included in a user interface associated with an active document (for example, see figure 1). As described above, these GUI elements may be configured to publish data to a specific data target. It is interpreted that input specifying the data target may be received during development of the active document.

In reference to claims 29, 30, and 32, the GUI elements taught by the combination of Risberg and Nawaz are configured by entering a data target into a posting page provided to the user, whereby as shown above, the GUI elements resultantly publish data to this data target. Thus these GUI elements are automatically configured without user programming and without the user input specifying source code.

With respect to claim 40, the combination of Risberg and Nawaz described above presents a method whereby a user may configure a GUI element to publish data to a particular data target. In particular, the combination teaches receiving user input during development of a

Art Unit: 2173

first computer program, namely an active document, the user input specifying a data target with which to associate the GUI element. As shown above, this user input does not include input specifying source code for the active document. Risberg additionally teaches that information entered by the user may be published (see column 3, line 66 – column 4, line 4). It is understood that such user input is entered into a GUI element. Moreover, it is understood that such user input may change the data associated with a GUI element. Consequently, the combination of Risberg and Nawaz further teach configuring the active document to receive user input changing data associated with the GUI element and publish the changed data to the specified data target during program execution.

In concern to claim 41, the combination of Risberg and Nawaz teach that the data target may be the same as the data source, as is shown above in the rejection for claims 18 and 19.

Regarding claim 42, Risberg teaches a method like that of claim 33, wherein a user enters a data source into a dialog box. In response, a GUI element is automatically configured to receive and display data from this data source, as is shown above. Risberg however does not explicitly disclose that this input specifying a data source is a URL, as is expressed in claim 33.

Like Risberg, Nawaz presents a method for configuring a GUI element, specifically a ticker, to receive and display data from a specific data source. With further similarity to the teachings of Risberg, this data may comprise financial data, namely stock prices (see column 3, lines 27-29). Regarding the claimed invention, Nawaz teaches that the data is received and displayed in the ticker from sources specified by URLs (see column 12, lines 23-40).

It would have therefore been obvious to one of ordinary skill in the art, having the teachings of Risberg and Nawaz before him at the time the invention was made, to modify the

Art Unit: 2173

method taught by Risberg such that the data sources are specified by URLs, as is taught by Nawaz. One would have been motivated to create such a combination because, as is demonstrated by Nawaz, URLs provide well-known and commonly used identification means for identifying data on a network.

In reference to claim 43, the application disclosed by Risberg involves displaying a GUI element associated with a first program, specifically an active document, wherein as shown above in the rejection for claim 7, the GUI element is displayed in response to user input received during development of the active document. Moreover, as shown above in the rejection for claim 15, Risberg as modified by the teachings of Nawaz, further teaches receiving user input during development of the active document, the user input specifying a data target with which to associate the GUI element, and in response, automatically configuring the active document to publish data associated with the GUI element to the specified data target during program execution.

The ideas expressed in claim 44 are also expressed in claim 32. Consequently, claim 44 is rejected for the reasons cited above in the rejection for claim 32.

With respect to claim 45, Risberg teaches that information entered by the user may be published (see column 3, line 66 – column 4, line 4). It is interpreted that such user input is entered into a GUI element, as is common in the art. Consequently, the combination of Risberg and Nawaz described above teaches publishing data, the published data comprising user input received by a GUI element during program execution.

As per claim 46, Risberg teaches that information in a constructed active document may be published (see column 3, line 66 – column 4, line 4). More specifically, Risberg discloses

Art Unit: 2173

that it is desirable to publish all or part of an active document (see column 2, lines 11-15).

Consequently, it is understood that the data displayed by a GUI element, which is part of an active document, may be published. As this data is displayed by the GUI element, the GUI element being part of a program, i.e. active document, it is further understood that the data is programmatically associated with the GUI element. Consequently, the combination of Risberg and Nawaz described above teaches publishing data, wherein the published data is programmatically associated with a GUI element.

Regarding claim 47, it is understood that when the active document taught by the combination of Risberg and Nawaz is executed, it is operable to publish data to a specified data target during execution of the active document, as is shown above in the rejection for claim 15.

Referring to claim 48, the active document taught by the combination of Risberg and Nawaz is considered a graphical program for the reasons presented above in the rejection for claim 10.

In regard to claim 49, it is understood that the method taught by the combination of Risberg and Nawaz is implemented via a "computer facility" (see column 2, lines 27-31 of Risberg). In other words, the method executes on a first computer. It is further understood, because of the reasons presented above in the rejection for claim 16, that the data target is a remote data target associated with a remote computer, i.e. second computer, wherein the first computer is operable to connect to the second computer, and wherein configuring the first computer program comprises configuring the first computer program to connect the remote data target and publish data to the remote data target during program execution.

As per claim 50, Risberg teaches that information entered by the user may be published (see column 3, line 66 – column 4, line 4). Information entered by the user is live data. Consequently, the combination of Risberg and Nawaz described above teaches publishing live data to a remote data target. Nawaz teaches that this information is received by the remote data target, wherein it may be displayed via a GUI element (see column 10, lines 16-28). It is inherent that this GUI element is associated with a computer program. Therefore, the combination of Risberg and Nawaz further teaches creating a second computer program operable to receive and display the live data from the remote data target.

In reference to claim 51, the combination of Risberg and Nawaz presents a method, whereby as shown above, a user may configure a GUI element to publish data to a particular data target. Nawaz particularly discloses that the data is transferred to the target computer via a server (see column 10, lines 20-22). Since the server is responsible for sending the data to the target computer, as is known in the art, it is understood that the data is simply sent to the server. The server then sends the data to the target computer. Consequently, the server is considered equivalent to the target. Thus, with the combination of Risberg and Nawaz described above, the data target is a server program.

Regarding claim 52, the combination of Risberg and Nawaz described above presents a method whereby a user may configure a GUI element to publish data to a specific data target. Additionally, and for the reasons disclosed above in the rejection for claims 1 and 7, Risberg teaches receiving user input during development of a first computer program, namely an active document, wherein the user input specifies a data source with which to associate the GUI element. As shown above, this user input does not specify source code for the active document.

Art Unit: 2173

In response to the reception of this user input, the GUI element is automatically configured to receive and display data from the specified data source. Consequently, it is understood that the combination of Risberg and Nawaz further teach a method like that recited in claim 52.

With respect to claim 53, the combination of Risberg and Nawaz teach that the data target may be the same as the data source, as is shown above in the rejection for claims 18 and 19.

As per claim 54, the combination of Risberg and Nawaz teach that a data source may be specified by a URL, as is shown above in the rejection for claim 42. Consequently, and for the reasons given above in the rejection for claim 42, it is interpreted that one of ordinary skill in the art would understand to similarly specify a data target by a URL.

With regard to claim 55, Risberg teaches displaying a GUI element on a display, and receiving user input specifying a data source with which to associate this GUI element, as is shown above in the rejection for claim 1. Similarly, Risberg as modified by Nawaz, further teaches receiving user input specifying a data target with which to associate a GUI element, wherein as described above, the GUI element is automatically configured to publish data associated with the GUI element to the specified data target. It is thus understood that with the method disclosed by Risberg and Nawaz, a user may specify both a data source and a data target with which to associate a GUI element. Moreover, it is understood that the specified data source may be the same as the specified data target. In such a case, the GUI element would be automatically configured to receive and display data from the remote data source, and publish data to the specified data target.

Referring to claim 56, the combination of Risberg and Nawaz teach a method, wherein as shown above in the rejection for claim 51, a GUI element of an active document being executed

Art Unit: 2173

is configured to write data to a server program. As shown above in the rejection for claim 50, this data may be live data. Consequently, Risberg and Nawaz teach executing a writer program, namely an active document, wherein the writer program is operable to write live data to a server program. Risberg further teaches that a reader program, specifically an active document, may also comprise a GUI element configured to receive and display data from a server, as is shown above in the rejection for claim 33. As shown above in the rejection for claim 3, the active document does not include user-specified source code for implementing the GUI element which receives the live data from the server program. Like the tickers disclosed by Nawaz, it is interpreted that the data published by a first active document may be received and displayed by a second active document. Thus the combination of Risberg and Nawaz teach a method comprising: executing a writer program, i.e. first active document, wherein the writer program is operable to write live data to a server program; executing a reader program, i.e. a second active document, where in the reader program includes a GUI element configured to subscribe to the data written to the server program; wherein the GUI element is operable to receive the live data from the server program and display the live data to a user of the reader program; and wherein the reader program does not include user-specified source code for implementing the GUI element receiving the live data from the server program.

Regarding claim 57, the GUI element, which is configured to receive and indicate live data from a server, indicates the live data by displaying the live data on the display screen, as is shown above.

With respect to claim 58, the combination of Risberg and Nawaz further teaches that the writer program, which is an active document, comprises a GUI element configured to publish

Art Unit: 2173

data to the server program. As shown above in the rejection for claim 51, the writer program writes live data to the server program by using this GUI element to publish data to the server program.

As per claim 59, the live data published by the writer program may comprise measurement data because of the reasons cited above in the rejection for claims 20 and 21.

Conclusion

The prior art made of record on form PTO-892 and not relied upon is considered pertinent to applicant's disclosure. Applicant is required under 37 C.F.R. § 1.111(c) to consider these references fully when responding to this action. The Heinzman U.S. Patent cited therein presents a method whereby an interface is used to send to or receive data from a remote device. The Peltier U.S. Patent cited therein presents a method for creating an interface to monitor and control a remote device. Lastly, the Ohara et al. U.S. Patent cited therein presents a method for creating a graphical program, the program comprising a plurality of connected nodes which visually indicate the functionality of the program.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Blaine Basom whose telephone number is (703) 305-7694. The examiner can normally be reached on Monday through Friday, from 8:30 am to 5:30 pm.

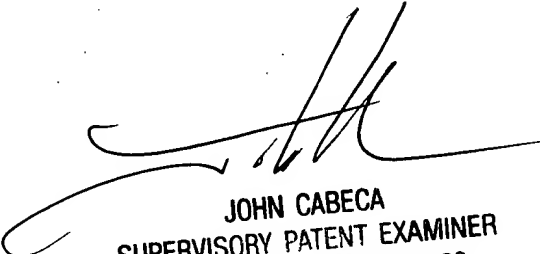
If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached on (703) 308-3116. The fax phone numbers for the

Art Unit: 2173

organization where this application or proceeding is assigned are (703) 746-7238 for regular communications and (703) 746-7240 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 305-3900.

btb
May 30, 2003



JOHN CABECA
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100